Dr. Lauer IT
00011010 00001100 01000111



Dr. Lauer IT

CANcorder

Benutzerhandbuch

CAN-Verkehr aufzeichnen, verstehen und dokumentieren

Produktübersicht und Betriebsanleitung

Version 1.3.1247

---

https://www.vanille.de/apps/cancorder/

Inhalt

CANcorder kennenlernen 3

CANcorder-Hilfe 4

Verbindung mit CAN-Hardware 5

Dateiformate 10

CAN-Frame-Tabelle 14

Transport-Nachrichten 16

Filterung 19

Protokollreferenz 22

Protokollhinweise 25

Plugins 29

J1939-Dashboard 34

Qualitätstest 35

Utilities 36

Fehlersuche 36

CANcorder kennenlernen

CANcorder macht aufgezeichneten Busverkehr nachvollziehbar: einzelne Frames, vollständige Diagnosenachrichten und ihre Bedeutung. Die Desktop-App richtet sich an Menschen, die Fahrzeugelektronik entwickeln, testen oder Fehler suchen und verstehen möchten, was auf einem CAN-Bus oder einer K-Line-Verbindung passiert ist.

Von der Aufzeichnung zur Erklärung

Öffne ein gespeichertes Protokoll oder verbinde unterstützte Hardware. Sieh dir die ursprünglichen Frames an, grenze die Ansicht mit Filtern ein und verfolge dann die zusammengesetzten Transportnachrichten und Protokollhinweise. Rohdaten und Interpretation stehen nah beieinander. So kannst du einen decodierten Wert anhand der zugehörigen Bytes überprüfen.

Typische Aufgaben sind die Suche nach einer Steuergeräteantwort, die Analyse einer Übertragung über mehrere Frames, der Vergleich von Antwortzeiten und die Beobachtung decodierter J1939-Werte. Mit Plugins kannst du eigene Interpretationen für herstellerspezifische Nachrichten ergänzen. Die folgenden Kapitel beschreiben die unterstützten Protokolle und ihre Grenzen.

Ergebnisse weitergeben

Speichere Aufzeichnungen für spätere Analysen oder erstelle einen PDF-Bericht mit ausgewählten Frames, Transportnachrichten, Protokollhinweisen und Antwortzeitstatistiken. Titel und Notizen helfen dabei, eine Aufzeichnung mit Kollegen zu besprechen oder einem Testprotokoll beizulegen. Unter Dateiformate findest du die verfügbaren Formate und Berichtsoptionen.

Den eigenen Arbeitsablauf ausprobieren

Zum Kennenlernen reicht eine vorhandene Logdatei oder das mitgelieferte Beispielprotokoll. Für die Offline-Analyse brauchst du keinen Adapter. Bei der Live-Aufzeichnung hängen die Möglichkeiten von Betriebssystem und Adapter ab: TCP-Logger, J2534-PassThru-Schnittstellen unter Windows, SocketCAN unter Linux und ausgewählte native USB-Adapter unter macOS. Prüfe vor der Hardwarewahl unter Verbindung mit Hardware die unterstützten Geräte, K-Line-Einschränkungen und Zugriffsmodi.

Zu diesem Handbuch

Dieses Handbuch beschreibt CANcorder 1.3.1247. Es enthält dieselben Kapitel wie die Hilfe in der App, einschließlich des Schnellstarts. Du kannst es von vorne lesen oder über das Inhaltsverzeichnis und die Kapitelverweise gezielt eine Aufgabe nachschlagen. Aktuelle Downloads und Informationen zur Lizenz findest du auf der CANcorder-Produktseite.

CANcorder-Hilfe

CANcorder ist ein CAN-Bus- und K-Line-Logger mit Analysefunktionen für Fahrzeugdiagnose. Die App verbindet sich per TCP, J2534 PassThru unter Windows, SocketCAN unter Linux oder mit ausgewählten USB-Adaptern unter macOS. J2534 unterstützt außerdem K-Line-Aufzeichnungen. CANcorder zeichnet CAN- und K-Line-Frames auf, setzt mehrteilige Nachrichten über ISO-TP, ISO-TP CEA, SAE TP2.0, K-Line TP, J1939, CCP und XCP zusammen und interpretiert UDS-, KWP- und OBD-II-Verkehr.

Schnellstart

1. **Mit Hardware verbinden**: Im Menü Verbindung einen gefundenen Logger auswählen oder IP/Port manuell eingeben
2. **Oder Logdatei laden**: Datei -> Protokoll öffnen verwenden oder eine Datei auf das Fenster ziehen
3. **Transport-Decoding aktivieren**: Im Filter-Panel "ISO-TP Decode" aktivieren, um zusammengesetzte Nachrichten zu sehen
4. **Hinweise aktivieren**: "Hints" einschalten, um Protokollinterpretationen zu sehen
5. **Beispielprotokoll laden**: Hilfe -> Beispielprotokoll laden verwenden, um OBD-II, UDS, KWP und Plugin-Hinweise auszuprobieren

Hilfethemen

Mit **PDF-Handbuch öffnen** in diesem Hilfefenster öffnest du das installierte Handbuch im PDF-Programm deines Systems. CANcorder enthält die deutsche und englische Ausgabe; der Button folgt der Sprache der App. Zum Lesen und Ausdrucken brauchst du keine Internetverbindung.

- Verbindung mit Hardware - TCP, J2534 (Windows, CAN + K-Line), SocketCAN (Linux), USB (macOS) und CAN-Frame-Versand
- Dateiformate - Unterstützte Logformate (CL1000, candump, candump CSV, ECUconnect, PCAPNG)
- CAN-Frame-Tabelle - Anzeige der Rohframes verstehen
- Transport-Nachrichten - Reassembly für ISO-TP, SAE TP2.0, K-Line TP, J1939, CCP und XCP
- Filterung - Nach ID, DLC, ECU und Nachrichtentyp filtern
- Protokollreferenz - Kurzzusammenfassung zu OBD-II, UDS, KWP2000, ISO-TP, J1939, CCP und XCP
- Protokollhinweise - Interpretation von UDS, OBD-II, KWP2000 und J1939
- Plugins - Protokollunterstützung mit eigenen Plugins erweitern
- J1939-Dashboard - Nach ECU gruppiertes Live-Dashboard für veränderliche J1939-Werte
- Qualitätstest - Empfangene, fehlende, doppelte und umsortierte Frames sowie Latenz und Jitter verstehen
- Utilities - Live-Scan-Helfer für macOS und Linux
- Fehlersuche - Tipps zu Verbindung, Decoding und Performance

Oberfläche

Die App besteht aus einer linken Seitenleiste und dem Hauptbereich für Analyse:

- **Verbindungs-Panel** - Verbindungsmodus, Adaptereinstellungen, Verbinden/Trennen, Recording-ID-Filter und Live-Statistiken
- **Qualitätstest-Panel** - Optionale Messung von Paketqualität
- **Filter-Panel** - ECU-, ID-, DLC- und Tabellenfilter
- **Transport-Panel** - Auswahl des Transport-Decoders, Adressierungsmodus, Hinweise und Smart Filtering
- **Hauptansicht** - CAN-Frame-Tabelle plus Transport-Nachrichten oder das J1939-Dashboard

Tastatur und Maus

- Frame/Nachricht anklicken: wählt den Eintrag aus und kopiert Payload-Daten in die Zwischenablage
- Shift+Klick auf Frame: setzt den Messpunkt für Delta-Zeit
- Datei auf das Fenster ziehen: öffnet die Logdatei
- Hinweiszeile anklicken: klappt Details ein oder aus
- Cmd/Ctrl+F: Hex-Suche öffnen
- Enter/Shift+Enter: zum nächsten/vorherigen Suchtreffer springen
- Escape: Suche schließen

Version

CANcorder 1.3.1247

Verbindung mit CAN-Hardware

CANcorder verbindet sich per TCP, J2534 PassThru unter Windows, SocketCAN unter Linux oder mit ausgewählten USB-Adaptern unter macOS. J2534 kann auch K-Line-Verkehr aufzeichnen.
TCP-Adapter müssen einen TCP-Server bereitstellen, der CAN-Frames im erwarteten binären Format streamt.

Verbindungs-Panel

Das Verbindungs-Panel ist der obere Bereich der linken Seitenleiste. Es enthält die Verbindungssteuerung, den Recording-ID-Filter und die Live-Statistiken.

Verbindungsmodi

- **TCP** (alle Plattformen) - Verbindung zu einem Logger oder Adapter, der Frames über TCP streamt.
- **J2534** (Windows) - installierte J2534-PassThru-DLLs verwenden. Unterstützt CAN und K-Line.
- **SocketCAN** (Linux) - lokale SocketCAN-Schnittstelle wie `can0` oder `vcan0` verwenden.
- **USB** (macOS) - gerätespezifische USB-Unterstützung. Aktuell: PEAK PCAN-USB FD, PCAN-USB Pro FD, Tactrix OpenPort 2.0, Rusoku TouCAN und gs_usb-kompatible Geräte.

Hinweis: J2534-, SocketCAN- und USB-Tabs erscheinen nur auf unterstützten Plattformen.

Felder (TCP)

Feld	Beschreibung	Default
Host	IP-Adresse oder Hostname des CAN-Adapters	leer
Port	TCP-Portnummer	leer

Zugriffsmodus (USB / SocketCAN / J2534)

Nicht-TCP-Modi bieten einen Zugriffsmodus:

- **Normal** - lesen und schreiben. Aktiviert nach dem Verbinden den Button zum Senden von Frames.
- **Listen-only** - Frames empfangen ohne zu senden. Der Adapter bestätigt keine Frames auf dem Bus und bleibt für andere Knoten unsichtbar. Senden ist nicht verfügbar.

Buttons

- **Verbinden** - Verbindung zum Adapter herstellen
- **Trennen** - Empfang stoppen und Verbindung schließen
- **Neu scannen** (J2534/USB) - erkannte Geräteliste aktualisieren
- **Frames senden** (grün) - Sendedialog öffnen (nur Normalmodus, nicht TCP)

Verbundener Zustand

Bei aktiver USB-, SocketCAN- oder J2534-Sitzung werden Gerät, Protokoll und Bitrate zu einer kompakten Zusammenfassung eingeklappt. Trennen und, im Normalmodus, Frames senden bleiben sichtbar.

Integrierte Statistiken

Unten im Verbindungs-Panel werden nach der ersten Verbindung Live-Statistiken angezeigt:

- **Frame-Rate** - aktuelle Empfangsrate in Frames pro Sekunde
- **Buslast** - sichtbar, wenn der Adapter genug Timingdaten für eine Schätzung liefert
- **Frames gesamt** - empfangene Frames in der aktuellen Sitzung
- **Empfangene Daten** - ungefähre Menge der Rohdaten
- **Streaming-Status** - animierter grüner Indikator, solange Live-Daten eintreffen

Statusanzeigen

- **Verbunden mit [host]:[port]** - aktive Verbindung
- **Fehlermeldung (rot)** - Verbindung fehlgeschlagen

CAN-Controller-Zustand

Der Abschnitt **CAN-Controller** zeigt `error-active`, `error-warning`, `error-passive`, `bus-off`, `stopped` oder `sleeping` getrennt vom Verbindungsstatus. Ein verbundener Adapter kann trotzdem bus-off sein. Verfügbare Zähler umfassen TEC/REC, beobachtete Fehlermeldungen, ACK- und Protokollfehler, Bus-off-Wechsel sowie Queue-Überläufe und verworfene Frames. Fehlende Zähler werden ausgelassen. Bei veralteten Daten bleibt der letzte Zustand ausdrücklich als veraltet sichtbar.

Der CAN-Empfang wird getrennt vom Controllerzustand angezeigt. Bis zum ersten empfangenen CAN-Frame bleibt er unbestätigt: Ein stiller Bus und ein fehlender physischer Busanschluss sind dann noch nicht unterscheidbar. Prüfe Busanschluss, ausgewählten Kanal, Bitrate und aktive Gegenstelle.

SocketCAN, gs_usb, PEAK und TouCAN liefern die Messwerte, die ihr Treiber und ihre Firmware unterstützen. Virtuelles CAN, OpenPort, J2534 und der reine TCP-Logger-Stream liefern keinen physischen Controller-Zustand. Die Werte beschreiben den angeschlossenen Controller, nicht jeden Bus-Teilnehmer. Sie messen keine Terminierung; `error-active` allein beweist keine erfolgreiche Kommunikation. Prüfe bei zwei Nodes die Bitrate, einen 120-Ω-Abschluss an jedem Ende und einen bestätigenden Teilnehmer im normalen Modus. Listen-only bestätigt keine Frames mit ACK.

J2534 PassThru (Windows)

Der Geräteauswahldialog zeigt pro Gerät:

- **Name** und **Hersteller**
- **Architektur-Badge** (32-bit / 64-bit)
- **API-Version** (v04.04 / v05.00), v5.0 in grün
- **Unterstützte Protokolle** als farbige Labels (CAN, ISO-TP, ISO 9141, ISO 14230, CAN-FD usw.)
- **Verfügbarkeit** - nicht verfügbare Geräte, etwa wegen fehlender DLL, sind ausgegraut

Nach der Auswahl des Geräts wählt man Protokoll (CAN oder K-Line) und Bitrate.

CANcorder durchsucht J2534-Registry-Einträge für v4.04 und v5.00 und lädt Treiber-DLLs über eine ausgelagerte Bridge. Dadurch funktionieren 32-bit- und 64-bit-Treiber, und ein Treiberabsturz beendet nur den Bridge-Prozess, nicht CANcorder.

CAN-Protokoll

Der Kanal wird mit Pass-all-Filter und `CAN_ID_BOTH` geöffnet, daher werden Standard-Frames (11-bit) und Extended-Frames (29-bit) empfangen.

K-Line-Protokoll

Im Protokoll-Dropdown **K-Line** wählen, um K-Line-Seriellbusverkehr zu erfassen. K-Line-TP-Decoding wird beim Verbinden automatisch aktiviert. Die Default-Bitrate ist 10400 (ISO 14230). Keepalive aktivieren, wenn die ECU periodische TesterPresent-Nachrichten für die Diagnosesitzung erwartet.

Unter macOS ist native OpenPort-K-Line-Unterstützung bus-initialisiert: CANcorder kann Roh-K-Line-Frames lesen und schreiben, nachdem ein anderes Tool, eine ECU-Wakeup-Sequenz oder ein Bench-Setup die K-Line-Sitzung initialisiert hat. Fast init, Five-baud init, automatische Protokollerkennung und Keepalive werden nur für Backends angeboten, die sie explizit unterstützen, etwa Windows J2534.

SocketCAN (Linux)

Im Tab **SocketCAN** eine Schnittstelle wie `can0` oder `vcan0` wählen und die Bitrate setzen. CANcorder verwendet den System-SocketCAN-Stack.

- Wenn keine Schnittstellen erscheinen, eine Schnittstelle mit dem gewohnten `ip link`/`can-utils`-Workflow anlegen oder aktivieren
- Die Bitrate sollte zur Schnittstellenkonfiguration passen

USB-Adapter (macOS)

Im Tab **USB** einen unterstützten Adapter auswählen. Aktuelle Unterstützung:

- **PEAK PCAN-USB FD** - einkanaliger CAN-FD-Adapter (IPEH-004022)
- **PEAK PCAN-USB Pro FD** - zweikanaliger CAN-FD-Adapter (IPEH-004061). Beide CAN-Kanäle erscheinen als getrennte Geräte. CAN 1 nutzt DB-9 Pins 7/2 (CANH/CANL), CAN 2 Pins 4/9.
- **Tactrix OpenPort 2.0** - klassisches CAN und natives K-Line unter macOS. Native K-Line ist für Sniffing/Schreiben auf bereits initialisiertem ISO-14230-Bus bei 10400 Baud getestet; K-Line-Initialisierung bleibt Windows-J2534-Funktion.
- **Rusoku TouCAN**
- **gs_usb-kompatible Geräte** - CANable, CANable 2.0, Innomaker USB2CAN, candleLight, CANtact und andere Adapter mit candleLight/gs_usb-Firmware

Weitere Hinweise:

- benötigt libusb (`brew install libusb`)
- verbinden und Bitrate wählen; CAN-Frames erscheinen wie bei TCP in der Tabelle
- für CAN-FD-fähige gs_usb-Adapter **CAN-FD** wählen und nominale sowie Daten-Bitrate setzen, z.B. `500000` / `2000000`

CAN-Frames senden

Bei USB (macOS), SocketCAN (Linux) oder J2534 (Windows) im **Normalmodus** erscheint ein grüner Button **Frames senden**. TCP-Verbindungen sind nur empfangend und unterstützen kein Senden.

Gesendete Frames werden in die Frame-Tabelle zurückgespiegelt, damit Requests und Antworten zusammen sichtbar sind.

Hinweis (J2534): Der Zeitstempel eines gesendeten Frames kann 0.000 anzeigen, wenn noch kein Frame empfangen wurde. J2534 liefert Zeitstempel nur für empfangene Frames und bietet keine API für die aktuelle Geräteuhr. Nach dem ersten empfangenen Frame stimmen die Zeitstempel später gesendeter Frames.

Einzel-Tab

Einzelne CAN-Frames interaktiv senden.

Feld	Beschreibung
CAN ID (hex)	Arbitration ID, z.B. 7DF
Std / Ext	Standard (11-bit) oder Extended (29-bit)
Data (hex bytes)	Payload-Bytes, z.B. 02 01 00
Pad to DLC	Mit 0 oder eigenem Byte bis DLC auffüllen
CAN-FD frame	Als FD-Frame senden, falls unterstützt

Nach dem Senden erscheint der Frame in der **Historie** (letzte 20 Einträge). Das Wiederholen-Symbol sendet einen früheren Frame erneut.

Sequenz-Tab

Mehrere Frames automatisch in einem von drei Modi senden:

- **Repeat** - denselben Frame N-mal mit konfigurierter Rate senden
- **Sweep** - CAN-ID-Bereich durchlaufen (Start -> Ende, mit Schrittweite) und feste Payload senden
- **Custom List** - eine Zeile pro Frame (`ID DATA`, z.B. `7DF 02 01 00`), optional in Endlosschleife

Alle Sequenzmodi verwenden dieselbe Ratensteuerung:

- **FPS-Modus** - Ziel-Frames pro Sekunde, z.B. 100 fps -> 10 ms Intervall
- **Intervallmodus** - Millisekunden zwischen Frames, z.B. 10 ms -> 100 fps

Ein Fortschrittsbalken zeigt den Sendeverlauf. **Stop** bricht eine laufende Sequenz ab.

Einstellungen

Alle Sendedialog-Einstellungen (IDs, Daten, Padding, Rate, Sequenzmodus, eigene Liste und Historie) werden gespeichert und beim nächsten Start wiederhergestellt.

Zeroconf-Discovery (mDNS/Bonjour, TCP)

CANcorder findet ECUconnect-Logger automatisch, wenn sie den Dienst `_ecuconnect-log._tcp.local.` per Zeroconf (mDNS/Bonjour) anbieten.

- Gefundene Logger erscheinen im Verbindungsmenü und im Verbindungsdialog
- Auswahl füllt Host/Port aus und verbindet direkt
- Wenn nichts gefunden wird, Host und Port manuell eingeben

Hinweis: Manche Netzwerke blockieren multicast DNS. Wenn Discovery keinen Logger findet, prüfen, ob mDNS am Logger aktiv ist und im Netzwerk nicht gefiltert wird.

Live-Streaming

Bei aktiver Verbindung empfängt CANcorder CAN-Frames in Echtzeit:

- Der Header zeigt "Recording" mit rotem Indikator
- Frames erscheinen beim Eintreffen in der CAN-Frame-Tabelle
- Das Statistikfeld zeigt Live-Frame-Rate und Summen
- Auto-Scroll hält die neuesten Frames sichtbar

Pause/Fortsetzen

Beim Live-Streaming erscheint ein Pause-Button:

- **Gelb (Pause)** - Auto-Scroll stoppen und durch die Historie scrollen
- **Grün (Play)** - Auto-Scroll zu den neuesten Frames fortsetzen

Pausieren stoppt nicht den Empfang; Frames werden weiter aufgezeichnet.

Frame-Speicherung

CANcorder hält empfangene Frames für schnellen Zugriff im Speicher. Lange Sitzungen können entsprechend viel RAM verbrauchen. Für längere Aufzeichnungen ist später ein Stream auf Disk oder ein Rolling Buffer sinnvoll.

Trennen

Beim Trennen mit aufgezeichneten Daten erscheint eine Rückfrage, um versehentlichen Datenverlust zu verhindern.

Dateien bei aktiver Verbindung laden

Wenn bei aktiver Verbindung eine Logdatei geöffnet wird, fragt CANcorder nach, ob die Verbindung getrennt und die Datei geladen werden soll.

Wire-Protokoll

CANcorder erwartet Frames in diesem binären Format:

Bytes	Feld	Format
0-7	Timestamp	64-bit big-endian Mikrosekunden
8-11	CAN ID	32-bit big-endian
12	Flags	bit0=EXT, bit1=FD, bit2=BRS, bit3=ESI
13	DLC	Datenlänge (0-64)
14+	Data	Rohbytes

Legacy-Sender, die Byte 12 nur auf `0` (standard) oder `1` (extended) setzen, werden weiterhin unterstützt.

Dateiformate

CANcorder kann Logdateien in mehreren Formaten importieren. Das Format wird automatisch anhand des Dateiinhalts erkannt.

Dateien öffnen

- **Menü**: Datei -> Protokoll öffnen
- **Toolbar**: Ordner-Symbol anklicken
- **Drag & Drop**: Datei auf das Fenster ziehen

Unterstützte Erweiterungen: `.txt`, `.log`, `.pcapng`, `.csv`

Ein geöffnetes Log wird nach einer externen Änderung oder Ersetzung automatisch neu geladen, sobald die Datei stabil ist. Anzeigefilter bleiben aktiv. Fehlt die Datei vorübergehend oder lässt sie sich nicht einlesen, bleibt das bisherige Log mit einem Statushinweis sichtbar; CANcorder versucht es automatisch erneut. Während einer Live-Aufzeichnung pausiert das Neuladen.

PDF-Berichte

Wähle **Datei** → **Als PDF exportieren …** (Cmd/Ctrl+Alt+E) oder den PDF-Button in der Werkzeugleiste. Wähle **Messages** für rohe Bus-Frames oder **Transport** für dekodierte Nachrichten und anschließend **Aktuelle Filter** oder **Alles**. Transport verwendet die aktuellen Decoder- und Hinweis-Einstellungen. Aktiviere zuerst die Transportdekodierung, falls noch keine dekodierten Nachrichten vorliegen.

„Aktuelle Filter“ verwendet dieselben ID-, ECU-, Längen- und Transportfilter wie die jeweilige Tabelle. Suchmarkierungen und ausgewählte Zeilen schränken den Export nicht ein. Der Bericht übernimmt den Stand des Logs und der Filter beim Öffnen des Dialogs; die Live-Aufzeichnung kann weiterlaufen.

Das Farb-PDF enthält die Auswahl, Originalnummern, Zeitangaben mit Mikrosekundenauflösung, vollständige Hex-/ASCII-Payloads und Seitenzahlen. Lange Payloads werden über mehrere Seiten fortgesetzt. Barlow und Iosevka sind eingebettet, damit der Bericht auch auf anderen Computern gleich aussieht. Der Export funktioniert offline und lässt sich während der Erstellung abbrechen.

Der Bericht verwendet A4-Hochformat. Die Präambel bekommt eine eigene Seite mit dem CANcorder-Icon neben dem Titel; die Tabelle beginnt immer auf einer neuen Seite. Lange Payloads zeigen acht Bytes pro Zeile mit Offset und ASCII. Die Fußzeile enthält die Version, bei lizenzierter Nutzung die Lizenz-E-Mail sowie Copyright und Website. Farbige Symbole ersetzen den Request-/Response-Statustext: Rechtspfeil für Anfragen, Häkchen für positive Antworten, Kreuz für negative Antworten, Uhr für ausstehende Antworten und Linkspfeil für sonstige Antworten. Die Präambel enthält eine beschriftete Legende. Kompakte Zeilen halten Adresspaare zusammen; Payloads setzen sich farblich von der Interpretation ab.

Die Antwortzeitstatistik zeigt Bestwert, Mittelwert, schlechtesten Wert, Jitter (Standardabweichung der Grundgesamtheit) und P95 sowie bei mehreren Teilnehmern eine Aufschlüsselung nach Steuergerät. Es zählen vollständige, eindeutig zuordenbare OBD-/UDS-/KWP-Paare, deren Request und Response beide exportiert werden. Gemessen wird vom letzten Request-Frame bis zum ersten Frame der finalen Antwort. Fehlen rohe Request-Frames, gilt einheitlich der Request-Beginn; der Bericht nennt die Messbasis. NRC 0x78 hält den Request offen. Ohne aufgezeichnete Zeitstempel, etwa bei minimalem candump CSV, gibt es keine Antwortzeitstatistik.

CL1000-Format

Das CL1000-Format wird von CAN-Loggern von CSS Electronics verwendet. CANcorder nutzt es für Import und Export.

Struktur

```
# Logger type: CAN-Logger-App
# HW rev: 1.0
# FW rev: 1.0
# Logger ID: app0001
# Time: 20231015T143052
# Bit-rate: 500000
Timestamp;Type;ID;Data
01T000000000;0;7E0;0322F190
01T000000050;0;7E8;1014622F190574F
```

Felder

Spalte	Beschreibung
Timestamp	DDTHHMMSSmmm - Tage, Stunden, Minuten, Sekunden, Millisek.
Type	0 = standard (11-bit), 1 = extended (29-bit)
ID	CAN-ID in Hex ohne Prefix
Data	Payload-Bytes in Hex, zusammenhängend

Angezeigte Metadaten

- Logger-Typ und ID
- Hardware-/Firmware-Revision
- Aufnahmezeit
- Bitrate
- Session- und Split-Nummern
- Silent-/Cyclic-Flags

candump-Format

Das candump-Format wird von Linux can-utils erzeugt (`candump -L`).

```
(1766060076.353543) can1 08FF0001#0200FFFFCF00AC7D
(1766060076.353755) can1 08FF0101#0100A25F000091B2
```

CAN-FD-Frames verwenden ein zweites # gefolgt von einem Flags-Nibble (Bit 0 = BRS, Bit 1 = ESI). CANcorder liest diese Form und markiert den Frame entsprechend:

```
(1766060076.353543) can1 123##1AABBCCDD
```

ECUmulator++ schreibt dieses Format mit `--can-log` und kodiert die Richtung über die Interface-Spalte: `<kanal>rx` für Frames in die Emulation hinein, `<kanal>tx` für Frames heraus.

Teil	Beschreibung
(timestamp)	Sekunden.Mikrosekunden; kann ohne RTC auch negativ sein
interface	CAN-Schnittstelle, z.B. can0 oder vcan0
id	CAN-ID in Hex; mehr als 3 Stellen = Extended Frame
#	Trennzeichen
data	Payload-Bytes in Hex, zusammenhängend

Angezeigt werden Schnittstellenname, Frame-Anzahl, Start-/Endzeit und Dauer.

Minimales candump CSV

Einige Tools exportieren CSV-Dateien mit nur Arbitration-ID und Payload-Bytes.

```
ID,Data Bytes
000007C0,06 06 FF 02 05 FF FF FF
00000130,F3 FF FF FF FF
```

Da keine Zeitstempel enthalten sind, erzeugt CANcorder monoton steigende Zeitstempel für die Anzeige. Start-/Endzeit und Dauer werden nicht angezeigt.

ECUconnect-Format

Logs aus der ECUconnect-iOS-App enthalten ISO-TP-Nachrichten mit Metadaten.

```
#META:timestamp=12:34:56.789
#META:vin=WBADP5555E123456
#META:ecu=DME
00:00:00.000,7E0,0322F190
00:00:00.050,7E8,1014622F190574F
```

Angezeigt werden VIN, ECU-Name, App-/Adapterinformationen, Tester- und ECU-Adressen, Taskbeschreibung sowie Nachrichten- und Fehlerzähler.

Besonderheit: ECUconnect-Logs werden wie PCAPNG-Dateien behandelt. Es wird nur die Transport-Tabelle angezeigt; ISO-TP-Decoding und Hinweise werden automatisch aktiviert.

PCAPNG-Format

PCAPNG-Dateien enthalten DoIP- oder HSFZ-Ethernet-Verkehr.

Unterstützte Protokolle:

- **HSFZ** - ISO-13400-2-Ethernet-Header-Framing
- **DoIP** - ISO 13400-1 Diagnostic over IP

Angezeigt werden Protokolltyp, VIN falls vorhanden, Paket- und Nachrichtenanzahl, Start-/Endzeit, IP-Adressen, ECU-Adressen und Netzwerkschnittstellen.

Besonderheit: PCAPNG wird aus Performancegründen im nativen Rust-Backend geparst. Angezeigt werden nur Transport-Nachrichten; ISO-TP-Decoding und Hinweise werden automatisch aktiviert.

Metadaten-Dialog

Das Info-Symbol neben dem geladenen Dateinamen öffnet die Metadaten:

- formatspezifischer Header mit farbigem Badge
- VIN-Anzeige für ECUconnect/PCAPNG
- Raster mit Metadatenfeldern
- Adresslisten und Schnittstelleninformationen

Formaterkennung

CANcorder erkennt Formate in dieser Reihenfolge:

1. **PCAPNG** - über `.pcapng`-Erweiterung
2. **ECUconnect** - über `#META:` im Inhalt
3. **Minimales candump CSV** - über `ID,Data Bytes` oder `id,data`
4. **candump** - über Muster `(timestamp) interface id#data`
5. **CL1000** - Fallback für andere Textdateien

CAN-Frame-Tabelle

Die CAN-Frame-Tabelle zeigt rohe CAN-Frames so, wie sie empfangen oder aus einer Datei geladen wurden.

Spalten

Spalte	Beschreibung
#	Frame-Index, beginnend bei 1
Time	vergangene Zeit seit dem ersten Frame in Sekunden
ID	CAN-ID in Hex
Ext	J für Extended (29-bit), N für Standard (11-bit)
DLC	Data Length Code, Anzahl Payload-Bytes
Data	Payload-Bytes in Hex, mit Leerzeichen
ASCII	druckbare ASCII-Darstellung, . für andere Bytes

Farben

ID-Spalte

- **Grün** - Standard-Frame (11-bit ID, 0x000-0x7FF)
- **Orange** - Extended-Frame (29-bit ID)

Zeilenauswahl

- **Blauer Balken** - aktuell ausgewählter Frame
- **Smaragdfarbener Balken** - Messpunkt für Zeitdifferenz

Interaktionen

Klick

- wählt den Frame aus
- kopiert Frame-Daten als Hex in die Zwischenablage, z.B. `03 22 F1 90`
- wählt bei aktivem Transport-Decoding die zugehörige Transport-Nachricht aus

Shift+Klick

- setzt einen Messpunkt für Delta-Zeit
- die Zeitspalte zeigt `Delta t`, also den Abstand zum Messpunkt
- normaler Klick löscht den Messpunkt

Scrollen

- Mausrad scrollt durch Frames
- bei Live-Streaming mit Auto-Scroll erscheinen neue Frames unten
- Pause erlaubt freies Scrollen durch die Historie

Frame-Zähler

Der Header zeigt `1,234 / 5,000 Frames`.

- erste Zahl: Frames passend zum aktuellen Filter
- zweite Zahl: alle geladenen Frames

Wenn nichts passt, erscheint "Keine Frames passen zum Filter".

Zeitanzeige

Normalmodus

Zeigt die vergangene Zeit seit dem ersten Frame:

- `0.000` - erster Frame
- `1.234` - 1,234 Sekunden nach dem ersten Frame

Delta-Modus (Shift+Klick)

Zeigt die Zeitdifferenz zum Messpunkt:

- `Delta t: 0.050` - 50 ms nach dem Messpunkt
- `Delta t: -0.100` - 100 ms vor dem Messpunkt

Geteilte Ansicht

Wenn Transport-Decoding aktiv ist, wird die Anzeige geteilt: oben CAN-Frames, unten Transport-Nachrichten. Der Trenner ist ziehbar.

- Trenner ziehen, um das Verhältnis zu ändern (15%-85%)
- der Trenner wird beim Hover oder Ziehen hervorgehoben

Performance

Die Tabelle nutzt virtualisiertes Scrollen für große Datenmengen:

- nur sichtbare Zeilen werden gerendert
- flüssiges Scrollen auch bei 10.000+ Frames
- Speicherverbrauch steigt mit der Anzahl im RAM gehaltener Frames
- Disk-basierte Aufzeichnung oder Rolling Buffer können lange Sitzungen entlasten

Transport-Nachrichten

ISO-TP (ISO 15765-2) ist ein Transportprotokoll, mit dem Nachrichten größer als 8 Byte über CAN übertragen werden können. CANcorder setzt solche mehrteiligen Nachrichten automatisch zusammen und unterstützt zusätzlich SAE TP2.0, K-Line TP, J1939, CCP und XCP.

Transport-Decoding aktivieren

Im Filter-Panel **ISO-TP Decode** aktivieren, um:

- mehrteilige Nachrichten zusammenzusetzen
- die Transport-Nachrichtentabelle anzuzeigen
- Protokollhinweise zu aktivieren

Adressierungsmodi

Der Modus wird im Dropdown des Filter-Panels gewählt:

Modus	Beschreibung
ISO-TP	Standard-ISO-TP mit 11-bit- oder 29-bit-CAN-IDs
ISO-TP (CEA)	erstes Datenbyte als Extended Address
SAE TP2.0	Volkswagen Group TP 2.0
K-Line TP	K-Line-Seriellbus-Transport
J1939	SAE-J1939-Transport (BAM und RTS/CTS)
CCP	CAN Calibration Protocol
XCP	Universal Measurement and Calibration Protocol on CAN

Extended Addressing (CEA) wird verwendet, wenn mehrere ECUs dieselbe CAN-ID teilen. K-Line TP wird automatisch gewählt, wenn per J2534 mit K-Line verbunden wird.

J1939-Transport

J1939 nutzt 29-bit Extended CAN-IDs mit Priorität, PGN, Source Address und Destination Address. CANcorder decodiert:

- **BAM (Broadcast Announce Message)** - Broadcast ohne Handshake
- **RTS/CTS (Request to Send / Clear to Send)** - Punkt-zu-Punkt-Uebertragung mit Flow Control
- **TP.CM / TP.DT** - Session-Steuerung und Datenframes

Fragmentierte J1939-Payloads werden als einzelne Nachrichten mit PGN-, Prioritäts- und Adressmetadaten angezeigt.

CCP

CCP wird für ECU-Kalibrierung und Messungen über CAN genutzt. Kommunikation erfolgt über feste CAN-IDs mit Command/Response-Muster:

- **CRO** - Tester-zu-ECU-Kommandos
- **DTO** - ECU-zu-Tester-Antworten inklusive Fehlercodes und Daten
- **DAQ** - zyklische Messdatenpakete von der ECU

CANcorder paart Kommandos und Antworten über das CTR-Byte und zeigt dekodierte Kommandonamen und Fehlercodes.

XCP

XCP ist der Nachfolger von CCP mit höherem Durchsatz und mehr Funktionen. Auf CAN nutzt es ebenfalls feste CAN-IDs:

- **CMD** - Tester-zu-ECU-Kommandos
- **RES/ERR** - positive Antworten oder Fehler
- **DAQ/STIM** - Mess- und Stimulationsdaten

CANcorder paart XCP-Kommandos und Antworten mit Timing-Heuristiken und zeigt Kommandonamen und Fehlercodes.

ISO-TP-Frame-Typen

Typ	Name	Zweck
SF	Single Frame	Nachricht passt in einen Frame (<=7 Bytes)
FF	First Frame	erster Frame einer mehrteiligen Nachricht
CF	Consecutive Frame	Fortsetzung einer mehrteiligen Nachricht
FC	Flow Control	Antwort des Empfängers auf FF

Spalten der Nachrichtentabelle

Spalte	Beschreibung
#	Nachrichtenindex, beginnend bei 1
Time	vergangene Zeit in Sekunden
TX ID	sendende CAN-ID
RX ID	empfangende CAN-ID
CEA	Extended-Address-Byte oder -
Dir	Richtung: REQ oder RSP
Len	gesamte Nachrichtenlänge in Bytes
Data	Payload in Hex

Richtungserkennung

CANcorder erkennt Request und Response über:

- **OBD-II-Muster** - IDs 0x7E0-0x7E7 als Requests, 0x7E8-0x7EF als Responses
- **UDS-Muster** - Response-ID = Request-ID + 8
- **Request-SIDs** - Nachrichten mit bekannten Request-Service-IDs

Datenanzeige

Kurze Nachrichten bis 40 Bytes werden inline angezeigt. Längere Nachrichten werden mit einem Aufklappindikator gekürzt; ein Klick zeigt alle Bytes.

Zeilenfarben

Farbe	Bedeutung
hellblauer Hintergrund	Request-Nachricht
hellgrüner Hintergrund	Response-Nachricht
dunkleres Blau	ausgewählt

Nachrichtenzähler

Der Header zeigt `234 / 1,000 Nachrichten`:

- erste Zahl: Nachrichten passend zum aktuellen Filter
- zweite Zahl: alle Nachrichten

Nachrichten auswählen

Ein Klick wählt die Nachricht aus, kopiert die Payload als Hex und hebt, falls CAN-Frames sichtbar sind, den ersten zugehörigen Frame hervor. Wird ein CAN-Frame ausgewählt, sucht CANcorder die Transport-Nachricht, die diesen Frame enthält. Das funktioniert auch für Flow-Control-Frames.

TransferData-Export

Bei UDS-TransferData-Sequenzen (0x36) erscheint **Transferdaten speichern**:

- zeigt Richtung (Download/Upload)
- zeigt Byte-Anzahl
- zeigt Zieladresse, falls vorhanden
- speichert die zusammengesetzten Binärdaten in eine Datei

Filterung

Das Filter-Panel hilft, bestimmte Frames und Nachrichten nach ID, Typ, Länge und weiteren Kriterien zu fokussieren.

Hex-Suche

Das Lupensymbol im Header öffnet die Suche nach Hex-Sequenzen in Frame- oder Nachrichtendaten.

Verwendung

1. Lupensymbol anklicken; es wird blau, wenn die Suche aktiv ist
2. Hex-Sequenz eingeben, z.B. `22F190`
3. Mit Auf/Ab-Buttons oder Enter/Shift+Enter durch Treffer springen
4. Escape schließt die Suche

Funktionen

- **Trefferzähler** - Anzeige im Format "3 von 15"
- **Hervorhebung** - aktueller Treffer mit amberfarbigem Ring, weitere Treffer dezent markiert
- **Bereichsbewusst** - sucht nur in sichtbaren Tabellen
- **Flexible Eingabe** - Leerzeichen und Trenner werden ignoriert (`22 F1 90` = `22F190` = `22:F1:90`)

ECU-Filter

Erkannte ECUs erscheinen oben im Filter-Panel als anklickbare Pills.

Wenige ECUs (<=6)

Jede ECU erscheint als eigene Pill:

- **Sichtbar** - farbiger Hintergrund
- **Ausgeblendet** - grau/dezent
- Anklicken schaltet die Sichtbarkeit um

Viele ECUs (>6)

Ein Dropdown zeigt:

- Buttons **Alle anzeigen** und **Alle ausblenden**
- Checkbox pro ECU
- Nachrichtenanzahl pro ECU

Was zählt als ECU?

Bei CAN-Frames wird jede eindeutige CAN-ID als ECU behandelt. Bei PCAPNG-/ECUconnect-Dateien werden TX- und RX-IDs gezählt.

ID-Filter

CAN-IDs werden in das Feld **ID Filter (hex)** eingegeben.

Muster	Treffer
7E0	exakte ID 0x7E0
7E0,7E8	IDs 0x7E0 oder 0x7E8
7E*	IDs beginnend mit 0x7E (0x7E0-0x7EF)
18DA*	Extended IDs beginnend mit 0x18DA

Für fortgeschrittene Filter können Regex-Muster verwendet werden.

Frame-Typ

Checkbox	Wirkung
Standard	11-bit CAN-IDs anzeigen
Extended	29-bit CAN-IDs anzeigen

Wenn beide deaktiviert werden, sind keine Frames sichtbar.

DLC-Bereich

Nach Data Length Code filtern:

Feld	Beschreibung	Bereich
Min DLC	minimale Payload-Länge	0-64
Max DLC	maximale Payload-Länge	0-64

Beispiele:

- Min=8, Max=8 zeigt nur 8-Byte-Frames
- Min=1 blendet leere Frames aus

Transport-Optionen

Steuerung	Beschreibung
ISO-TP Decode	mehrteilige Nachrichten zusammensetzen
Protocol	ISO-TP, ISO-TP (CEA), SAE TP2.0, K-Line TP, J1939, CCP oder XCP

Details stehen unter Transport-Nachrichten.

Hinweis-Optionen

Steuerung	Beschreibung
Hints (OBD2/UDS/KWP)	Protokollinterpretation anzeigen
Smart filtering	auf Request/Response-Paare derselben ECU filtern

Details stehen unter Protokollhinweise.

Hinweis-Typfilter

Bei aktivierten Hinweisen erscheinen Filter-Pills:

Pill	Farbe	Bedeutung
Request	Blau	UDS-/OBD-II-Requests
Positive	Grün	erfolgreiche Antworten
NRC	Rot	Negative Response Codes
Busy	Orange	Server busy (0x21 NRC)

Pills anklicken, um Typen ein- oder auszublenden.

Filterkombinationen

Filter werden mit UND-Logik kombiniert:

- Ein Frame muss alle aktiven Filter erfüllen
- ECU-Filter UND ID-Filter UND DLC-Bereich UND Frame-Typ

Beispiel für DME-Antworten:

1. ID-Filter auf `7E8` setzen
2. Nur "Standard" aktivieren
3. Hints aktivieren und nur Typ "Positive" anzeigen

Zähleranzeige

Beide Tabellen zeigen gefilterte und gesamte Anzahl:

```
CAN Frames: 1,234 / 5,000 Frames
ISO-TP:     234 / 1,000 Nachrichten
```

Wenn nichts passt: "Keine Frames passen zum Filter" oder "Keine Nachrichten passen zum Filter".

Filter zurücksetzen

1. Im ECU-Dropdown "Alle anzeigen" wählen
2. ID-Filter leeren
3. Standard und Extended aktivieren
4. DLC-Bereich auf 0-64 setzen
5. Alle Hinweis-Typen anzeigen

Protokollreferenz

Diese Seite ist ein praktischer Spickzettel für häufige Fahrzeugdiagnoseprotokolle in CANcorder. Sie ersetzt keine vollständige Spezifikation.

ISO-TP-Grundlagen (ISO 15765-2)

- ISO-TP setzt mehrteilige Payloads zusammen, die in CAN-Frames transportiert werden
- Single Frame (SF) enthält bis zu 7 Bytes, bei Extended Addressing bis zu 6 Bytes
- First Frame (FF) startet eine mehrteilige Uebertragung, Consecutive Frames (CF) setzen sie fort
- Flow-Control-Frames (FC) werden für das Decoding in CANcorder ignoriert

Beispiel Single Frame:

```
CAN Data: 02 10 01
          ^^
          Laenge = 2 Bytes
Payload: 10 01
```

Beispiel Multi-Frame-Payload mit Länge 10:

```
FF: 10 0A 22 F1 90 01 02 03
CF: 21 04 05 06 07
Payload: 22 F1 90 01 02 03 04 05 06 07
```

UDS-Service-IDs (ISO 14229)

Häufige Service-IDs:

```
SID  Name                        Bedeutung
0x10 DiagnosticSessionControl    Diagnosesitzung wechseln
0x11 ECUReset                    ECU zuruecksetzen
0x22 ReadDataByIdentifier        DID-Werte lesen, z.B. VIN
0x23 ReadMemoryByAddress         Speicherbloecke lesen
0x2E WriteDataByIdentifier       DID-Werte schreiben
0x2F InputOutputControlById      IO-Steuerung
0x31 RoutineControl              Routinen starten/stoppen
0x34 RequestDownload             Download starten
0x35 RequestUpload               Upload starten
0x36 TransferData                Datenbloecke uebertragen
0x37 RequestTransferExit         Transfer beenden
0x3E TesterPresent               Sitzung offen halten
0x7F NegativeResponse            Fehlerantwort mit NRC
```

Positive Antworten echoen typischerweise die SID plus 0x40:

```
Request 0x22 -> positive Antwort 0x62
Request 0x10 -> positive Antwort 0x50
```

OBD-II (SAE J1979)

- funktionale Request-ID: 0x7DF (Broadcast)
- ECU-Response-IDs: 0x7E8-0x7EF
- Service $01 = Current Data, $09 = Vehicle Information

Häufige PID-Beispiele:

```
Service  PID  Bedeutung
0x01     0C   Motordrehzahl
0x01     0D   Fahrzeuggeschwindigkeit
0x01     05   Kuehlmitteltemperatur
0x09     02   VIN
```

KWP2000 (ISO 14230)

- älter als UDS, aber in vielen ECUs weiterhin zu finden
- Services überschneiden sich mit UDS, verwenden aber andere IDs
- CANcorder-Hinweise decodieren best-effort; herstellerspezifische Details brauchen oft ein Plugin

SAE TP2.0 (VAG TP 2.0)

- Volkswagen-Group-TP2.0 ist ein eigenes Transportprotokoll auf 11-bit-CAN, keine ISO-TP-Variante
- Kanäle werden vor der Datenübertragung dynamisch ausgehandelt
- Payloads laufen danach in TPDU-Frames mit ACK- und Retransmit-Semantik
- CANcorder setzt TP2.0-Payloads zusammen und wendet anschließend UDS-/KWP-Hinweise auf die dekodierten Bytes an
- typische feste CAN-IDs für Channel-Setup liegen im Bereich `0x200` bis `0x2EF`; dynamische Datenkanäle werden beim Setup vergeben

K-Line (ISO 14230 / ISO 9141)

- K-Line ist ein Eindraht-Seriellbus für Diagnose in älteren Fahrzeugen
- typischerweise 10400 Baud, auch wenn einzelne Implementierungen abweichen
- CANcorder erfasst K-Line-Verkehr über J2534-PassThru-Geräte mit ISO9141-/ISO14230-Unterstützung
- K-Line-TP-Decoding setzt Request-/Response-Nachrichten zusammen und nutzt UDS-/KWP-Hinweise
- Keepalive sendet periodische TesterPresent-Nachrichten, um die Diagnosesitzung offen zu halten

J1939 (SAE J1939)

- Standard für Nutzfahrzeuge mit 29-bit Extended CAN-IDs
- CAN-ID codiert Priorität, Data Page, PF, PS und Source Address
- PGN (Parameter Group Number) identifiziert den Nachrichtentyp
- zwei Transportmodi für Payloads größer als 8 Bytes:

```
Modus    Control PGN   Data PGN   Beschreibung
BAM      0x00EC00      0x00EB00   Broadcast ohne Handshake
RTS/CTS  0x00EC00      0x00EB00   Punkt-zu-Punkt mit Flow Control
```

TP.CM-Control-Byte-Werte:

```
Byte  Name      Richtung
0x10  RTS       Request, Initiator
0x11  CTS       Response, Empfaenger gibt frei
0x13  EOM_ACK   Response, Transfer beendet
0x20  BAM       Broadcast-Ankuendigung
0xFF  ABORT     Transfer abbrechen
```

CANcorder filtert ISO-15765-3-Verkehr (EDP=1 oder PF=0xDA/0xDB) aus, damit UDS-Diagnoseframes nicht als J1939 fehlklassifiziert werden.

CCP

- ASAM MCD-1 CCP für ECU-Kalibrierung und Messung
- festes CAN-ID-Paar: eine ID für Kommandos (CRO), eine für Antworten (DTO)
- Kommandos verwenden ein Counter-Byte (CTR) zur Request/Response-Zuordnung

Häufige Kommandos:

Code	Name	Zweck
0x01	CONNECT	Sitzung herstellen
0x02	SET_MTA	Speichertransferadresse setzen
0x03	DOWNLOAD	in ECU-Speicher schreiben
0x04	UPLOAD	aus ECU-Speicher lesen
0x06	START_STOP	DAQ-Listen starten/stoppen
0x07	DISCONNECT	Sitzung beenden
0x0F	SHORT_UP	kurzer Upload bis 5 Bytes
0x12	GET_SEED	Security Access Seed
0x13	UNLOCK	Security Access Key

XCP

- ASAM MCD-1 XCP, Nachfolger von CCP mit höherem Durchsatz
- auf CAN: festes ID-Paar für CMD/RES plus optionale DAQ-/STIM-IDs
- Kommandos verwenden PID-Bytes im Bereich 0xFF..0xC0 absteigend

Häufige Kommandos:

Code	Name	Zweck
0xFF	CONNECT	Sitzung herstellen
0xFE	DISCONNECT	Sitzung beenden
0xFD	GET_STATUS	Sitzungsstatus abfragen
0xF5	UPLOAD	aus ECU-Speicher lesen
0xF4	SHORT_UPLOAD	kurzer Upload
0xF0	DOWNLOAD	in ECU-Speicher schreiben
0xED	SHORT_DOWNLOAD	kurzer Download
0xE2	SET_DAQ_PTR	DAQ-Zeiger konfigurieren
0xE1	WRITE_DAQ	DAQ-Listeneintrag schreiben
0xDE	START_STOP_DAQ_LIST	DAQ-Liste starten/stoppen
0xDD	START_STOP_SYNCH	synchron starten/stoppen

Request/Response-Muster

- 11-bit-Adressierung: Requests meist 0x7E0-0x7E7, Responses 0x7E8-0x7EF
- 29-bit-Adressierung: oft 0x18DAxxYY, wobei Source-/Target-Bytes zwischen Request und Response tauschen

Protokollhinweise

Protokollhinweise interpretieren zusammengesetzte Transport-Payloads nach Fahrzeugdiagnosestandards wie UDS, OBD-II, KWP2000 und J1939. CANcorder unterstützt Hinweise für ISO-TP, ISO-TP (CEA), SAE TP2.0, K-Line TP und J1939.

Hinweise aktivieren

1. Im Filter-Panel **Transport Decode** aktivieren
2. Transportprotokoll wählen: **ISO-TP**, **ISO-TP (CEA)**, **SAE TP2.0**, **K-Line TP** oder **J1939**
3. **Hints (OBD2/UDS/KWP)** aktivieren

Hinweiszeilen erscheinen unter interpretierten Nachrichten in der Transport-Tabelle.

Bei **SAE TP2.0** zeigt CANcorder auch ausgewählte Transport-Control-Frames mit Hinweisen, darunter Channel Setup/Ack, Connection Setup/Ack, Connection Test, Break und Disconnect.

Bei **J1939** zeigt CANcorder PGN-Namen für direkte und zusammengesetzte Payloads, dekodiert ausgewählte öffentliche SPN-Formeln häufiger PGNs und stellt ASCII-artige Identifikationspayloads wie Software Identification, Component Identification, Vehicle Identification und ECU Identification dar, wenn möglich.

J1939-Kataloge

CANcorder bringt eine kleine eingebaute J1939-Referenz aus öffentlichem/Open-Source-Material mit. Für breitere Abdeckung wird ein externes `J1939db.json` gesucht und über den eingebauten Katalog gemerged.

Suchorte:

- `AppConfig/J1939db.json`
- `AppConfig/j1939/J1939db.json`
- `~/.config/cancorder/J1939db.json`
- `~/.config/pretty_j1939/J1939db.json`

Kompatibel sind Dateien im `J1939db.json`-Format, wie sie Tools wie `pretty_j1939` und `j1939decode` verwenden.

Anzeige

Hinweise erscheinen als gelbe Zeilen unter Nachrichten:

```
Hinweis  Request  ReadDataByIdentifier: F190 (VIN)
```

Teil	Bedeutung
Indikator	markiert Hinweiszeile
Statusicon	Request/Response-Typ
Servicename	interpretierter Service
Details	Parameter und Werte

Statusicons

Typ	Bedeutung	Farbe
Request	Anfrage	Blau
Positive response	erfolgreiche Antwort	Grün
Negative response	negative Antwort	Rot
Busy	Busy / Request wiederholen	Orange

Hinweise aufklappen

Ein Klick auf eine Hinweiszeile öffnet Details. Die Detailtabelle zeigt:

- **Feldname** - was die Bytes bedeuten
- **Wert** - interpretierter Wert
- **Rohbytes** - originale Hexdaten

Beispiel:

Feld	Wert	Rohbytes
Diagnostic Service	ReadDataByIdentifier	22
Data Identifier	F190 (VIN)	F1 90
VIN	WBADP5555E123456	57 42 ...

UDS-Services

CANcorder interpretiert diese UDS-Services (ISO 14229):

SID	Service	Beschreibung
0x10	DiagnosticSessionControl	Diagnosesitzung wechseln
0x11	ECUReset	ECU zurücksetzen
0x14	ClearDiagnosticInformation	DTCs löschen
0x19	ReadDTCInformation	Fehlercodes lesen
0x22	ReadDataByIdentifier	ECU-Daten lesen
0x27	SecurityAccess	geschützte Funktionen
0x2E	WriteDataByIdentifier	ECU-Daten schreiben
0x34	RequestDownload	Speicherdownload starten
0x35	RequestUpload	Speicherupload starten
0x36	TransferData	Speicherblöcke übertragen
0x37	RequestTransferExit	Transfer beenden
0x3E	TesterPresent	Sitzung offen halten

Diagnosesitzungen (0x10)

Session	Name
0x01	Default Session
0x02	Programming Session
0x03	Extended Diagnostic Session
0x04	Safety System Diagnostic Session

Häufige Data Identifier (0x22/0x2E)

DID	Beschreibung
0xF186	aktive Diagnosesitzung
0xF187	Ersatzteilnummer des Fahrzeugherstellers
0xF188	ECU-Softwarenummer
0xF189	ECU-Softwareversion
0xF18A	System Supplier Identifier
0xF18C	ECU-Seriennummer
0xF190	VIN
0xF197	Systemname oder Motortyp

OBD-II-Services

Für OBD-II-Diagnose-Requests (IDs 0x7DF, 0x7E0-0x7E7):

Mode	Beschreibung
0x01	aktuelle Daten / Live-PIDs
0x02	Freeze-Frame-Daten
0x03	gespeicherte DTCs
0x04	DTCs und Freeze Frame löschen
0x05	Lambdasondenüberwachung
0x06	On-board-Monitoring-Testergebnis
0x07	pending DTCs
0x08	On-board-Systeme steuern
0x09	Fahrzeuginformationen
0x0A	permanente DTCs

Negative Responses

Wenn eine ECU einen Request ablehnt, antwortet sie mit Service 0x7F.

Häufige NRC-Codes:

Code	Name	Bedeutung
0x10	generalReject	Request abgelehnt
0x11	serviceNotSupported	Service nicht verfügbar
0x12	subFunctionNotSupported	Subfunktion nicht verfügbar
0x13	incorrectMessageLength	falsche Datenlänge
0x14	responseTooLong	Antwort überschreitet Puffer
0x21	busyRepeatRequest	ECU beschäftigt
0x22	conditionsNotCorrect	Voraussetzungen fehlen
0x24	requestSequenceError	falsche Request-Reihenfolge
0x31	requestOutOfRange	Parameter außerhalb Bereich
0x33	securityAccessDenied	Security nicht freigeschaltet
0x35	invalidKey	falscher Security-Key
0x36	exceededNumberOfAttempts	zu viele Fehlversuche
0x72	generalProgrammingFailure	Flash-Schreiben fehlgeschl.
0x78	responsePending	Verarbeitung läuft

Smart Filtering

Wenn **Smart filtering** aktiv ist:

- werden nur Hinweise für ECU-Paare mit Request und Response gezeigt
- bleiben Hinweise für kanonische Diagnose-Request-IDs wie `7DF`, `7E0-7E7`, `18DA*`, `18DB*` sichtbar, auch ohne Antwort
- werden mehrdeutige Broadcast-Nachrichten ohne Antwort ausgefiltert
- wird Rauschen in vollen CAN-Logs reduziert
- können bei **ISO-TP (CEA)** Hinweise erst nach der ersten passenden Antwort erscheinen
- entstehen bei **SAE TP2.0** Hinweise aus der zusammengesetzten TPDU-Payload nach dem TP2.0-Decoding

Die Funktion ist für Live-Streams und CAN-Frame-Logs verfügbar, nicht für PCAPNG/ECUconnect.

Hinweis-Typfilter

Die farbigen Pills im Filter-Panel zeigen oder verbergen:

- **Request** (blau) - ausgehende Diagnose-Requests
- **Positive** (grün) - erfolgreiche Antworten
- **NRC** (rot) - Negative Response Codes

- **Busy** (orange) - 0x21 busyRepeatRequest

Eigene Protokolle

Für herstellerspezifische Protokolle siehe Plugins.

Plugins

Plugins erweitern das Hinweis-System von CANcorder um herstellerspezifische Diagnoseprotokolle.

Plugin-Speicherorte

Ort	Zweck
App-Store-Build: AppConfig/plugins	Benutzerplugins, sandboxed
Direktdownload: ~/.config/cancorder/plugins/	Benutzerplugins als JSON
Built-in	mit der App gebündelt

App-Store-Builds enthalten ein Beispielplugin (DynoTune Example Protocol) als Referenz.

Plugins anzeigen

Das Menü **Plugins** zeigt:

- **Geladene Plugins** - grünes Puzzle-Symbol, Klick für Details
- **Fehlgeschlagene Plugins** - rotes X, Klick zeigt den Fehler

Plugin-Details

Ein Klick auf ein Plugin zeigt:

- Titel und Beschreibung
- Plugin-ID und Version
- Dateipfad
- definierte Services und Subfunktionen

Plugin-Struktur

Plugins sind JSON-Dateien:

```
{
  "id": "my-protocol",
  "name": "My Protocol",
  "version": "1.0",
  "title": "My Custom Protocol",
  "subtitle": "For XYZ ECUs",
  "description": "Decodes proprietary diagnostic messages",
  "services": { "...": "..." }
}
```

Pflichtfelder

Feld	Beschreibung
id	eindeutige Kennung
name	interner Name
version	Plugin-Version
services	Service-Definitionen

Optionale Felder

Feld	Beschreibung
title	Anzeigename in der UI
subtitle	Untertitel
description	ausführliche Beschreibung
carrierService	UDS-Service als Wrapper
prefixes	alternative Nachrichtenprefixe
negativeResponseCodes	eigene NRC-Bedeutungen

Services definieren

```
{
  "services": {
    "62": {
      "name": "CustomRead",
      "subfunctions": {
        "48": {
          "name": "ReadVersion",
          "requestFields": [],
          "responseFields": []
        }
      }
    }
  }
}
```

Service-Eigenschaften

Eigenschaft	Beschreibung
name	Servicename für Anzeige
subfunctions	Map von Subfunction-ID zu Definition

Subfunction-Eigenschaften

Eigenschaft	Beschreibung
name	Subfunction-Name
requestFields	Felder in Request-Nachricht
responseFields	Felder in Response-Nachricht

Felddefinitionen

```
{
  "name": "Software Version",
  "offset": 1,
  "length": 2,
  "format": "version",
  "bigEndian": true
}
```

Feldeigenschaften

Eigenschaft	Beschreibung	Default
name	Anzeigename	Pflicht
offset	Byte-Offset nach SID	Pflicht
length	Anzahl Bytes	Pflicht
format	Werteformat	hex
bigEndian	Byte-Reihenfolge	true

Formattypen

Format	Beschreibung	Beispiel
hex	Hexadezimal	4F0A
decimal	Dezimalzahl	1024
ascii	ASCII-String	ECU123
version	Major.minor	2.5
address	Speicheradresse	0x80001234

Carrier Services

Manche Protokolle laufen getunnelt über Standard-UDS-Services. `carrierService` gibt den Wrapper an:

```
{
  "carrierService": 62,
  "services": {
    "62": {
      "subfunctions": {
        "48": { "name": "CustomCommand" }
      }
    }
  }
}
```

Das teilt CANcorder mit, dass Service 0x3E (62) eigene Subfunktionen transportiert.

Alternative Prefixe

Wenn ein Protokoll über mehrere UDS-Services erreichbar ist:

```
{
  "prefixes": [
    {
      "bytes": [34, 255, 255],
      "description": "Via ReadDataByIdentifier 0xFFFF"
    }
  ]
}
```

Conversation Tracking

Wenn ein Plugin einen Request erkennt, merkt CANcorder sich das für das ECU-Paar (TX ID <-> RX ID). Spätere Antworten dieses ECU-Paars bekommt das Plugin zuerst zum Decoding.

So bleibt die Decodierung innerhalb einer Diagnosesitzung konsistent.

Eingebaute Plugins

VehiCAL

Dekodiert das VehiCAL-Kalibrierprotokoll:

Subfunktion	Name	Zweck
0x30	Ident	ECU-Patchstatus prüfen
0x31	ReadMemoryByAddress	Speicher lesen (v1)
0x32	WriteMemoryByAddress	Speicher schreiben
0x33	ReadFastArray	Lesen per FastArray (v2)
0x35	ReadFastArrayInplace	Inline-FastArray lesen

Plugins erstellen

1. JSON-Datei in `~/.config/cancorder/plugins/` erstellen
2. Protokollstruktur definieren

3. CANcorder neu starten oder neu laden

Beispiel: einfaches Protokoll

```
{
  "id": "my-ecu",
  "name": "MyECU",
  "version": "1.0",
  "title": "My ECU Protocol",
  "services": {
    "34": {
      "name": "CustomService",
      "subfunctions": {
        "1": {
          "name": "GetStatus",
          "responseFields": [
            { "name": "Status", "offset": 0, "length": 1, "format": "hex" },
            { "name": "Counter", "offset": 1, "length": 2, "format": "decimal" }
          ]
        }
      }
    }
  }
}
```

Fehlersuche

Plugin wird nicht geladen

Im Plugins-Menü nach fehlgeschlagenen Plugins suchen. Häufige Ursachen:

- **JSON-Syntaxfehler** - JSON validieren
- **Pflichtfeld fehlt** - id, name, version und services prüfen
- **Doppelte ID** - Plugin-IDs müssen eindeutig sein

Hinweise erscheinen nicht

- ISO-TP Decode und Hints aktivieren
- prüfen, ob die Service-ID zur Nachricht passt
- falls verwendet, Subfunction-ID prüfen

J1939-Dashboard

Das J1939-Dashboard zeigt eine Echtzeitübersicht dekodierter J1939-Parameterwerte mit Sparkline-Verlauf, gruppiert nach Source-ECU.

Dashboard öffnen

1. Im Transport-Panel den Modus **J1939** wählen
2. **Transport-Nachrichten anzeigen** aktivieren
3. In der Header-Toolbar das **Dashboard-Symbol** anklicken

Das Dashboard ersetzt die geteilte Tabellenansicht. Ein erneuter Klick kehrt zu den Tabellen zurück.

Funktionsweise

CANcorder dekodiert jede J1939-Payload-Nachricht mit dem PGN-Katalog. Numerische Felder wie Drehzahl, Temperaturen, Drücke oder Spannungen werden pro Source-ECU und PGN verfolgt.

Angezeigt werden nur Felder, deren Wert sich **mindestens einmal geändert** hat. Statische Werte werden ausgeblendet, damit die Ansicht übersichtlich bleibt.

Dashboard-Karten

Jede Karte zeigt:

- PGN-Name und Label, z.B. EEC1 - Electronic Engine Controller 1
- Feldname, z.B. Engine Speed
- aktuellen Wert in großer Schrift mit farbcodierter Einheit
- Sparkline der letzten 120 Samples
- Min-/Max-Bereich unter der Sparkline

Werte werden pro Feld höchstens alle 50 ms gesampelt.

Farben

Einheit	Farbe
rpm	Orange
km/h	Grün
deg C / C	Cyan
%	Gelb
V	Lila
kPa	Rose
km	Teal
andere	Blau

ECU-Gruppen

Karten werden nach J1939-Source-Adresse gruppiert. Jede Gruppe zeigt:

- ECU-Name aus der J1939-Adresstabelle, z.B. "Engine #1"
- Hex-Adresse, z.B. 0x00
- Anzahl Felder

Gruppen werden nach Source-Adresse sortiert, Felder innerhalb der Gruppe nach PGN und Feldname.

PGN-Katalog

Das Dashboard dekodiert Felder aus dem eingebauten PGN-Katalog (EEC1, EEC2, ET1, CCVS1, VEP1, VD, EFL/P1) und aus externen Katalogen:

- `~/.config/cancorder/J1939db.json`
- `~/.config/pretty_j1939/J1939db.json`

Details zum Format stehen unter Protokollhinweise.

Tipps

- Das Dashboard verarbeitet Nachrichten inkrementell; nach dem Umschalten holt es bereits aufgezeichnete Daten nach
- Daten löschen setzt das Dashboard zurück
- Live-Aufzeichnungen und importierte Logdateien werden unterstützt
- Helles und dunkles Theme werden unterstützt

Qualitätstest

Das Qualitätstest-Panel prüft, ob ein CAN-Aufzeichnungspfad Frames verliert, dupliziert oder in anderer Reihenfolge liefert.

Messwerte

- **Received** - Anzahl eindeutiger Testframes im aktuellen Lauf
- **Missing** - fehlende Sequenznummern zwischen erstem und letztem empfangenem Testframe
- **Duplicates** - Frames mit bereits gesehener Sequenznummer
- **Reordered** - Frames, die nach einer neueren Sequenznummer eingetroffen sind

Latenz und Jitter

- **Relative Latency** - zusätzliche Empfangsverzögerung im Vergleich zum schnellsten Testframe des aktuellen Laufs. Das ist relativ, keine absolute Ende-zu-Ende-Buslatenz.
- **Jitter** - Schwankung der Paketabstände am Empfänger. Bei `1 fps` sollte der Abstand nahe `1000 ms` liegen; Jitter zeigt die Abweichung davon.

Neuen Test starten

Ein neuer Qualitätstest löscht die gesammelten Pakete des vorherigen Laufs, bevor die Messung beginnt. Wenn alte Ergebnisse noch sichtbar sind, fragt CANcorder vor dem Verwerfen nach.

Utilities

CANcorder hat begleitende Utilities für Live-Scanning und Aufzeichnung, die Daten in die App einspeisen.

macOS

- **GS_USB + RusokuCAN** - Live-Scanning unter macOS mit GS_USB-Adaptern und RusokuCAN.

Linux

- **SocketCAN** - Live-Scanning über SocketCAN-Adapter und zugehörige Tools.

Links

Die Utilities werden hier gepflegt:

https://github.com/Automotive-Swift/CANcorder

Fehlersuche

Verbindungsprobleme

- Im Menü Verbindung einen gefundenen Logger auswählen (Zeroconf/mDNS)
- Wenn keine Logger erscheinen, prüfen, ob Rechner und Logger im selben Netzwerk sind
- Bei manueller Verbindung IP-Adresse und Port des Loggers eingeben
- Erreichbarkeit des Loggers prüfen, z.B. gleiches Netzwerk oder USB-Tethering
- Wenn bereits eine Verbindung aktiv ist, vor dem Laden von Beispielprotokollen trennen

Keine Frames sichtbar

- Prüfen, ob der Logger Frames sendet; die Statusanzeige sollte verbunden sein
- Sicherstellen, dass Filter nicht alles ausblenden: ECUs zurücksetzen, ID-Filter leeren
- "Show Standard" und "Show Extended" umschalten, um den Adressierungsmodus zu prüfen

J2534-Probleme (Windows)

- **"Bridge executable not found"** - Die J2534-Bridge wurde nicht gebaut. `make bridge` oder `make dev` ausführen; `make dev` baut die Bridge automatisch.

- **Gerät sichtbar, verbindet aber nicht** - Adapter einstecken und Stromversorgung prüfen. Manche Adapter brauchen externe 12V.
- **Verbunden, aber keine Frames** - Bitrate muss zum CAN-Bus passen. CANcorder nutzt `CAN_ID_BOTH` für 11-bit- und 29-bit-Empfang.
- **Bridge process died** - Die J2534-DLL ist eventuell abgestürzt. Das ist von CANcorder isoliert; einfach neu verbinden. Wenn es wiederholt passiert, andere Treiberversion testen.
- **Diagnose-Logging** - Um alle Bridge-Requests nach stderr zu loggen, `J2534_BRIDGE_VERBOSE=1` setzen.

ISO-TP-Decoding bleibt leer

- ISO-TP-Decoding hängt von korrekter Adressierung ab (11-bit vs. 29-bit)
- Manche Logs enthalten nur rohe CAN-Frames ohne ISO-TP-Verkehr
- Hints einschalten, um zu prüfen, ob Nachrichten nach UDS/OBD/KWP aussehen

Plugin-Probleme

- Plugins werden beim Start aus `~/.config/cancorder/plugins/*.json` geladen
- Schemafehler oder ungültiges JSON verhindern das Laden; Konsolenlogs prüfen
- Wenn ein Plugin zu aggressiv matched, Matching-Regeln einschränken, etwa Service-IDs oder Subfunktionen

Performance-Tipps

- Sehr große Logs brauchen Zeit für Parsing und Rendering; warten, bis das Parsing abgeschlossen ist
- Bei Live-Aufzeichnung mit hoher Framerate Auto-Scroll pausieren oder Transport-Decoding ausblenden
- Filter mit komplexen Wildcards über Millionen Frames können langsam sein

Export-Probleme

- Export verwendet CL1000-Format (`Timestamp;Type;ID;Data`)
- Wenn Speichern fehlschlägt, Dateirechte und Zielpfad prüfen